

Unit: II
Lecture: 4
Design Heuristic and Design Document

heuristic means using experience to learn and improve

Fan in/ Fan out:

Fan-in is a **measure of the number of process or functions that call some other function or process** (say X).

Fan out is the number of functions that are known by the function X.

The fan out of a module is the number of its immediately subordinate modules.

The fan in of a module is the number of its immediately super-ordinate (boss) modules.

A high value for fan-in means that the X is tightly coupled to the rest of the design and alters to X will have extensive knock-on effects.

A high value for fan-out suggests that the overall difficulty of X may be high because of the difficulty of the control logic required to coordinate the called components.

Design Heuristics for effective modularity:

1. Evaluate the first iteration of the program structure to reduce coupling and improve cohesion.
2. Attempt to minimize structures with high fan-out, strive for fan-in as depth increases.
3. Keep the scope of effect of a module within the scope of control of that module.
 - The scope of effect of module e is defined as all other modules that are affected by a decision made in module e. The scope of control of module e is all modules that are subordinate to module e.
4. Evaluate module interfaces to reduce complexity and redundancy and improve consistency. Module interface complexity is a prime cause of software errors.
5. Define modules whose function is predictable, but avoid modules that are overly restrictive. A module is predictable when it can be treated as a black box i.e. the same external data will be produced regardless of internal processing details.
6. Strive for controlled entry modules by avoiding pathological connections. This design heuristic warns against content coupling. Software is easier to understand and therefore easier to maintain when module interfaces

are constrained and controlled. Pathological connection refers to branches or references into the middle of a module.

Design Documentation:

A software design document is a written report of a software product's design, describing its overall architecture.

A software design document helps to ensure the design specification of the software are understood and is clear to all. It specifies what is possible with the product and how it can be accomplished.

The Design Specification addresses different aspects of the design model and is completed as the designer refines his representation of the software.

- First, the scope of the design effort is described.
- Next, the data design is specified. Database structure, any external file structures, internal data structures and a cross reference that connects data objects to specific files are all defined.
- The architectural design indicates how the program architecture has been derived from the analysis model.
- The design of external and internal program interfaces is represented and a detailed design of the human/machine interface is described. In some cases, a detailed prototype of a GUI may be represented.
- Design constraints such as physical memory limitations or the necessity for a specialized external interface may dictate special requirements.
- The final section of the Design Specification contains supplementary data. Algorithm descriptions, alternative procedures, tabular data, excerpts from other documents and other relevant information are presented as a special note or as a separate appendix.